Classifying User Messages For Managing Web Forum Data

Sumit Bhatia[α], Prakhar Biyani[β] and Prasenjit Mitra[α,β]
[α]Department of Computer Science and Engineering
[β]College of Information Science and Technology
Pennsylvania State University
University Park, PA-16802, USA
sumit@cse.psu.edu, pbiyani@ist.psu.edu, pmitra@ist.psu.edu

ABSTRACT

Online discussion forums have become a popular medium for users to discuss with and seek information from other users having similar interests. A typical discussion thread consists of a sequence of posts posted by multiple users. All the posts in a thread are not equally useful and serve a different purpose providing different types of information (some posts contain questions, some answers, etc.). Identifying the purpose and nature of each post in a discussion thread is an interesting research problem as it can help in improving information extraction and intelligent assistance techniques [9]. We study the problem of classifying a given post as per its purpose in the discussion thread. We employ features based on the post's content, structure of the thread, behavior of the participating users and sentiment analysis of post's content. We achieve decent classification performance and also analyze the relative importance of different features used for the post classification task.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and Software – Question-answering (fact retrieval) systems; H.3.5 [**Information Storage and Retrieval**]: Online Information Services – Web-based services

General Terms

Human factors, Algorithms, Experimentation

Keywords

Online forums, message boards, classification, speech act classification, dialogue act classification.

1. INTRODUCTION

Online discussion forums have become quite popular as they provide an easily accessible platform to users in different parts of the World to come together and share information and discuss issues of common interest. Thousands of web forums devoted to a multitude of topics exist where millions of users regularly participate in various discussions. People use web forums to discuss and ask questions about various topics such as news, sports, technology, health, etc. The archives of web forums contain millions of such discussion threads and act as a valuable repository of human generated information that needs to be efficiently managed. There have been efforts to develop customized retrieval models for searching discussion threads [1, 6, 16], extracting useful information such as question-answer pairs from previous discussion threads [4, 5, 8] etc. In this paper, *we study the problem of classifying individual posts as per their role/purpose in the discussion thread.*

1.1 Why Post Classification?

A typical discussion thread consists of a number of individual *posts* or messages posted by different participating users. Often, the thread initiator posts a question and other users may offer possible solutions, ask for details, provide feedback about the proposed solutions etc. Identifying the purpose of each such post is essential for intelligent and effective utilization of the information contained in the thread. Some possible application scenarios are as follows:

- Systems for searching web forums can utilize this information for thread ranking. Threads containing solutions to a given problem can be assigned a higher weight than the threads that do not have an answer post. Likewise, threads that contain posts providing positive feedback about the solutions proposed in the thread can be ranked higher than the threads that have no feedback information or that contain negative feedback posts.

- Classifying forum posts according to their role can be utilized for assessing user roles in the discussions and improving information extraction and intelligent assistance techniques [9].

- Knowing the role and importance of different posts in a given thread is also useful for question answer detection algorithms as well as for summarizing a discussion thread. For example, a very concise summary of the thread can be constructed by using only the posts in which the question is being asked and the posts in which the solutions are being provided. Zhou and

Hovy [19] discuss challenges in summarizing dynamically created textual information (as is the case with online discussion forums) and argue that identifying the text segments (posts in case of forum threads) belonging to different categories (e.g. question, answers) is essential for creating effective summaries.

- Usually, threads in a web forum are displayed to users sorted by the time of posting of last message in the thread. Instead, an alternative scheme could be to present threads with unresolved questions first. This scheme can be useful especially for technical forums where people ask a lot of questions. Experienced users in the forum that generally provide answers to many questions [1] can then easily find threads with unanswered questions and provide the necessary information.

1.2 Our Contributions

We build upon the forum post classification task introduced in the Mailing List and Forum (MLAF) track at Forum for Information Retrieval Evaluation[1] (FIRE) and evaluate the effects of content based, structural, user based and sentiment based features for the post classification task. We achieve strong results using the proposed feature set. The dataset used for experiments in this paper is being made available for the research community[2].

The rest of this paper is organized as follows. Section 2 provides a survey of related literature. Section 3 describes the classes for posts in web forums and Section 4 describes the feature set used for post classification. Section 5 describes the dataset used in this work and results of classification experiments. Section 6 concludes the paper and provides directions for future research.

2. RELATED WORK

The most similar work to our work is that of dialogue act classification in natural language processing where the purpose is to classify different utterances according to their role or purpose in a conversation [12]. Dialogue act classification can be performed for spoken conversation (e.g. work by Stolke et al. [14]) as well as written conversation, the latter being similar in nature to our research. Cohen et al. [3] classify email messages according to the purpose of the email message in a business setting. They identified a set of *email verbs* (e.g. request, deliver, propose, commit etc.) and used text classification methods to detect if a given email message contains a specific email verb or not. Lampert et al. [11] propose a well-grounded set of definitions for *requests* and *commitments* in e-mail based on manual annotation experiments carried out with the Enron e-mail corpus. The work on dialogue act tagging for online forums by Kim et al. [10] is most similar to our work. Their work focuses on uncovering the thread content structure in the form of post-post linkages, i.e., identifying the (previous) posts in a thread to which a post responds and the type of relationship between the linked posts.

In addition to these directly related works, there also exist works dealing with the problem of knowledge extraction from forums. Yang *et al.,* utilize the linkages and relationships between pages in an online forum site to extract structured metadata like post title, post content etc [18]. Forums have also been used as a data source for question answering systems. Cong *et al.* propose techniques to extract question answer pairs from online forums[4]. Their question detection algorithm use sequential pattern features called labeled sequential patterns (LSPs) as features to distinguish between question and non-question sentences. Classical features like 5W1H words, presence of question mark are also used as features for question detection. Their answer extraction algorithm ranks the posts in a forum thread based on similarity with questions and user information to output a ranked list of candidate answers. Building upon this work and utilizing the same question detection approach, Ding *et al.* use conditional random fields to identify relationships between different posts in a thread to extract context and answers of the questions posed in a single thread[5]. Hong and Davison describe a classification based approach for detecting whether the first post of a thread is a question and then finding the potential answer post from the remaining posts in the thread [8]. A translation language model and query likelihood based retrieval model for question answer archives is proposed by Xue *et al.,* [17]. A general ranking framework for factual question answering is discussed by Bian *et al.,* [2]. Further, there also have been efforts to develop customized retrieval models for searching web forum data [1, 6].

3. DESCRIPTION OF CLASSES

We classify a given user post in a discussion thread into following eight classes as per its role in the discussion thread.

1. **Question:** The poster asks a question which initiates discussion in the thread. This is usually the first post in the thread but not always. Often, the topic initiator or some other user may ask other related questions in the thread.

2. **Repeat Question:** Some user repeats a previously asked question (e.g. *Me too having the same problem.*).

3. **Clarification:** The poster asks clarifying questions in order to gather more details about the problem/question being asked.

4. **Further Details:** The poster provides more details about the problem as asked by other fellow posters.

5. **Solution:** The poster suggests a solution to the problem being discussed in the thread.

6. **Positive Feedback:** Somebody tries the suggested solution and provides a positive feedback if the solution worked.

7. **Negative Feedback:** Somebody tries the suggested solution and provides a negative feedback if the solution did not work.

8. **Junk:** There is no useful information in the post. For example, someone justs posts a smiley or some comments that is unrelated to the topic being discussed. For example, *Sorry, I don't know how to solve this.*

[1] http://www.isical.ac.in/~fire/

[2] http://www.cse.psu.edu/~sub194/datasets/PostsWithLabels.tar.gz

Class	Post Content
Thread Title	Newly opened windows hide behind the Gnome panel :(
Question	When I open a new window, it opens at the top left corner. The top of the window hides behind the top panel so that I have to $< Alt + left >$ button to move the window first. I am using Compiz-Fusion in Gutsy. Does anybody know of a fix for this? Thank you!
Junk	anybody?
Solution	I'm pretty sure in the Compiz Manager there's a setting that changes where windows open on the screen. Place windows, I believe.
+ve Feedback	You're right! Thank you!!!

Table 1: An example thread with class labels for the posts. Note that the first row contains the thread title which is included to provide the context for the discussion going on in the thread. Thread title is not one of the target classes.

Table 1 shows an example thread with each post of the thread labeled with the appropriate class label. The first seven classes were described in the MLAF task at FIRE [3]. We added the eighth class as we observed that a significant number of posts in the threads do not contain any useful information and it is essential to identify such posts. Also note that even though the MLAF task provides a test dataset, we chose not to use it as it did not have the labels for the *junk* class. Further, the dataset provided a random set of user posts from discussion threads and their respective class labels. We however, wanted to experiment with certain user level and thread level features (see Section 4) and the thread level and user level information was not available in the provided dataset. Hence, we created our own dataset for experiments in this paper.

4. FEATURE DESCRIPTION

We use a variety of features for classifying forum messages into the eight classes as described above. Table 2 lists all the features used in this work and in the following subsections we describe the motivations behind using the said features.

4.1 Content Based Features

The content of the post should be a very good indicator of the nature of the post. For example, we expect the posts that answer the questions/issues raised in the initial post to have a relatively higher similarity with the title of the thread and the initial post. On the other hand, we expect that *off-topic* posts to have relatively low similarity scores. Based on these considerations, for each post we use as features the cosine similarity scores with the thread title, the initial post of the thread and the whole thread. In addition to content similarity, presence of question marks and any of the 5W1H question words hints that a question is being asked in the post. Likewise, if one of the previous posts is being quoted in the post[4], it is often the case that the user is responding to the quoted post.

4.2 User Features

Different users in a forum exhibit different message posting behavior. While some of the more experienced and knowledgeable users act as *information providers* and answer a lot of questions, there are a lot of users that mainly act as *information seekers* asking for solutions to their problems. Hence, intuitively the class of a post should have some dependency on the user. In order to capture this dependency, for each post we use as features the authority score of the poster [1], total number of messages posted by the user and if the user is the thread initiator.

4.3 Structural Features

We expect the location/position of the post in the thread to be an indicator of the class of the post. For example, ideally the problem being discussed is described in the first post of the thread and the clarifying questions and details are generally being discussed in first few posts whereas the posts containing the solutions and user feedback should be the last few posts of the thread. Hence we use the absolute and relative position of the post in thread as features. Similarly, we expect the posts where the problem and the solutions are being discussed to be longer than the posts where the feedback is provided. Hence, length of the post is also an important feature and we use four different versions of this feature (ref. Table 2). While computing these features, stemming was performed using Porter's stemmer [13] and stop words were removed using a general stop word list of 429 words used in the Onix Test Retrieval Toolkit[5].

4.4 Sentiment Features

These features try to capture the emotion/sentiment of the post. We expect the posts in which the users describe their problems and the posts where a negative feedback to a suggested solution is being provided to be of a negative tone. Likewise, the posts where users suggest a solution to the problem being discussed in the thread as well as the posts where a positive feedback is being provided should have a positive tone. We compute sentiment strength scores for each post using the SentiStrength algorithm as described by Thelwall et al. [15]. We use the implementation of the algorithm as available from http://sentistrength.wlv.ac.uk/. SentiStrength algorithm is specifically developed for sentiment detection and extracting sentiment strengths from short informal texts like forum posts, blog comments, etc. The method takes into account the grammar and spelling conventions (e.g. terms like LOL, OMG, bcoz etc.) that are common in cyberspace and computes positive and negative sentiment strength scores for a given piece of text using a set of rules and language patterns associated with the sentiment. In a given piece of text, there can be multiple word patterns/rules indicative of positive or negative sentiment. The software implementation computes two versions of positive and negative sentiment strength scores for each piece of text: *(i)* using the strongest indicative word patterns and *(ii)* us-

[3] http://www.isical.ac.in/~fire/

[4] Identified by "Quote" box in the threads used in this work. Different forum software provide different mechanisms to quote a post.

[5] http://www.lextek.com/manuals/onix/stopwords1.html

Feature Name	Description	Type
	Content Based Features	
IsQuote	Does the post quote a previous post? 1 if yes, 0 otherwise.	Binary
TitleSim	Cosine similarity between the post and thread title.	Real
InitSim	Cosine similarity between the post and first post of the thread.	Real
ThreadSim	Cosine similarity between the post and entire thread.	Real
QuestionMark	Does the post contain a question mark.	Binary
Duplicate	Does the post contain the keywords *same, similar*.	Binary
5W1H	Does the post contain a word from {what, where, when, why, who, how}.	Binary
	Structural Features	
AbsPos	Absolute position of a post in the thread	Numerical
NormPos	Normalized position of a post in the thread. Computed as (Absolute Position/Total number of posts in the thread)	Real
PostLength	Total number of words in a post after stopwords removal.	Numerical
PostLengthUnique	Unique number of words in a post after stopwords removal.	Numerical
PostLengthStemmed	Total number of words in a post after stopwords removal and stemming.	Numerical
PostLengthUniqueStemmed	Unique number of words in a post after stopwords removal and stemming.	Numerical
	User Features	
UserPostCount	Total number of messages posted by the poster.	Numeric
IsStarter	Is the post made by the topic starter? 1 if yes, 0 otherwise.	Binary
UserAuth	Authority score [1] of the poster.	Real
	Sentiment Based Features	
Thank	Does the post contain the keyword *thank*.	Binary
ExclamationMark	Does the post contain an exclamation mark.	Binary
–ve Feedback	Does the post contain the keywords *did not, does not*.	Binary
SentimentScore	Sentiment scores of the post as computed by SentiStrength [15]. (4 features, see text for detail).	Numeric

Table 2: Description of various features used for the classification task.

ing all the indicative word patterns and taking their average. Thus, we get four different sentiment strength scores for each post. In addition to sentiment strength scores, we also use the presence of emotion indicating punctuations like exclamation marks and presence of keywords like "thank" etc.

5. EXPERIMENTS

5.1 Data Description

For the experiments in this paper, we randomly sampled 100 threads from the dataset[6] of discussion threads from Ubuntu forums[7] used in previous research [1]. The threads consist of discussions related to the problems faced by various users of the Ubuntu operating system. There are a total of 556 posts in the 100 threads used in this work and the longest thread consists of 35 user posts.

In order to obtain class labels for the post, we recruited three human evaluators. All the three evaluators were senior year undergraduate students in computer science and were well versed with the Ubuntu operating system. The evaluators were provided with class definitions and were asked to assign the most suitable class label to each post. The evaluators worked independently of each other. The final class label of each post was then decided by the majority vote. Out of 556 posts in the dataset, a majority decision was achieved for 529 (95.14%) posts. The remaining 27 posts for which all the three evaluators assigned different class labels were discarded. Table 3 summarizes the distribution of different classes in our dataset. The dataset is available for download for research purposes (http://www.cse.psu.edu/~sub194/datasets/PostsWithLabels.tar.gz).

5.2 Experimental Protocol

We experimented with a variety of supervised machine learning algorithms like support vector machine, naive Bayes classifier, decision trees, multi-layer perceptron and the logit model classifier (logistic regression). The experiments were performed using the Weka data mining toolkit [7]. The performance of all the classifiers was comparable with the logit model achieving the best classification performance. Due to space constraints, we only report the results of experiments with the logit model classifier. We use stringent 10 folds cross validation and the results reported are averaged over the ten folds.

5.3 Forum Post Classification Results

[6]Can be downloaded from http://www.cse.psu.edu/~sub194/datasets/ForumData.tar.gz

[7]http://ubuntuforums.org

Class	Number of Instances
Question	134
Repeat Question	17
Clarification	29
Further Details	55
Solution	207
Negative Feedback	11
Positive Feedback	25
Junk	51
Total	529

Table 3: Distribution of different classes in the final dataset.

Classification Accuracy	72.02%
Precision	0.697
F1-Measure	0.700
Kappa Statistic	0.6168
Mean absolute error	0.0982
Root mean square error	0.2232

Table 4: Classification results.

Table 4 reports the results for forum post classification using the logit model classifier. We achieved an overall classification accuracy of 72.02% with a precision of 0.697 and F-1 measure of 0.700. Table 5 further summarizes the individual results for each class. We report precision, F-1 measure, true positive and false positive rates and the area under the ROC curve. From the table we observe that the classifier performance for the *Question* and *Suggest Solution* classes is relatively higher when compared with the other six classes with F-1 values of 0.856 and 0.810, respectively. Such high F-1 values for the *Question* and *Suggest Solution* classes is highly desirable as the posts corresponding to these two classes contain the most important information in the thread – the problem being discussed and its solution. Identifying such posts with high accuracy is crucial for applications like forum search, thread summarization, question-answer pair detection etc. We also note that the classifier performance for the *Clarification* and *Negative Feedback* classes is worst among the eight classes (F-1 values of 0.190 and 0.154, respectively). This poor performance can be attributed to the small number of posts belonging to these two categories in our dataset (29 posts out of 529 for *Clarification* and 11 posts out of 529 for *Negative Feedback* class) and thus, the inability of the classifiers to generalize over this small amount of data.

5.4 Relative Importance of Different Features

In this subsection, we investigate the effect of different types of features for the post classification task. We perform the classification experiment using only one type of feature at a time. Table 6 reports the classification results for each feature type. We report precision, F-1 measure and accuracy values. As before, ten folds cross validation was performed and results reported are averaged over the ten folds. We note that the highest individual performance is achieved by content based features and the lowest individual performance is achieved by sentiment features. We expect the content of a post to be a very strong indicator of the nature of the post and hence, the high performance of content based features that take into account relationship of the post content with the remaining thread. Further, while we expect the content based, user based and structural features to be helpful in identifying posts belonging to all the classes, sentiment features are expected to be most useful in identifying posts belonging to the two feedback classes. In addition, the small number of posts belonging to the two feedback classes (ref. Table 3) may also be responsible for the low performance of the sentiment based features.

Next, we study the importance of each individual feature for the post classification task. We evaluate all the features individually by measuring the chi-squared statistic with respect to the class label and rank all the features by their chi-square values. Table 7 lists the top ten features for the post classification task along with the minimum, maximum values of the features and the standard deviation of the respective feature values. We note that no sentiment based features appear in the list of top ten features in accord with previous observations.

Class	Precision	F-1 Measure	Accuracy
Content	0.469	0.514	59.36%
Structural	0.382	0.432	50.86%
User	0.392	0.471	59.55%
Sentiment	0.311	0.358	45.94%
All	0.697	0.700	72.02%

Table 6: Classification results obtained using content based, structural, user based and sentiment features individually.

6. CONCLUSIONS AND FUTURE WORK

In this work, we investigated the problem of classifying individual user posts in an online discussion thread. For post classification, we experimented with a variety of features derived from the post's content, thread structure, user behavior and sentiment analysis of the post's text. We achieved decent classification accuracy and per-class analysis revealed that the best performance was achieved for classes *question* and *suggest solution*. Further, analysis of individual features showed that the content based features were most useful for the post classification task and the performance of sentiment based features was worst among all the features studied. Our

Feature	Min.	Max.	Mean	StdDev.
InitPostSim	0	1	0.307	0.359
PostPosition	1	35	6.501	6.612
ThreadSim	0	0.993	0.470	0.254
IsStarter	0	1	0.452	0.498
LengthUnique	0	274	17.491	20.263
LengthUniqueStemmed	0	224	16.509	17.670
Length	0	428	21.456	31.070
UserAuthority	0	0.979	0.123	0.174
QuestionMark	0	1	0.374	0.484
TitleSim	0	0.913	0.162	0.193

Table 7: Top 10 features ranked by chi-square values.

Class	Precision	F-1 Measure	TP Rate	FP Rate	ROC Area
Question	0.847	0.856	0.866	0.053	0.959
Repeat Question	0.750	0.621	0.529	0.006	0.893
Clarification	0.308	0.190	0.138	0.018	0.920
Further Details	0.614	0.625	0.636	0.046	0.918
Suggest Solution	0.747	0.810	0.884	0.193	0.912
Negative Feedback	0.500	0.154	0.091	0.002	0.618
Positive Feedback	0.350	0.311	0.28	0.026	0.895
Noise	0.605	0.553	0.510	0.036	0.907
Overall	0.697	0.700	0.720	0.099	0.917

Table 5: Classification results for each class.

future work will focus on employing a larger set of features to improve classification performance. We also plan on using the class labels to improve summarization of discussion threads and thread retrieval.

7. REFERENCES

[1] S. Bhatia and P. Mitra. Adopting inference networks for online thread retrieval. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15*, pages 1300–1305, 2010.

[2] J. Bian, Y. Liu, E. Agichtein, and H. Zha. Finding the right facts in the crowd: factoid question answering over social media. In *WWW '08*, pages 467–476, New York, NY, USA, 2008. ACM.

[3] W. W. Cohen, V. R. Carvalho, and T. M. Mitchell. Learning to classify email into "speech acts". In *EMNLP*, pages 309–316. ACL, 2004.

[4] G. Cong, L. Wang, C.-Y. Lin, Y.-I. Song, and Y. Sun. Finding question-answer pairs from online forums. In S.-H. Myaeng, D. W. Oard, F. Sebastiani, T.-S. Chua, and M.-K. Leong, editors, *SIGIR*, pages 467–474. ACM, 2008.

[5] S. Ding, G. Cong, C. Lin, and X. Zhu. Using conditional random fields to extract contexts and answers of questions from online forums. In *Proceedings of ACL08 - HLT 2008*, 2008.

[6] J. L. Elsas and J. G. Carbonell. It pays to be picky: an evaluation of thread retrieval in online forums. In *SIGIR '09*, pages 714–715, 2009.

[7] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: An update. *SIGKDD Explorations*, 11(1), 2009.

[8] L. Hong and B. D. Davison. A classification-based approach to question answering in discussion boards. In *SIGIR '09*, pages 171–178, 2009.

[9] J. Kim, G. Chern, D. Feng, E. Shaw, and E. Hovy. Mining and assessing discussions on the web through speech act analysis. In *In Proceedings of the ISWC'06 Workshop on Web Content Mining with Human Language Technologies*, 2006.

[10] S. N. Kim, L. Wang, and T. Baldwin. Tagging and linking web forum posts. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, CoNLL '10, pages 192–202, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[11] A. Lampert, R. Dale, and C. Paris. The nature of requests and commitments in email messages. In *In Proceedings of the AAAI Workshop on Enhanced Messaging*, 2008.

[12] G. Murray, S. Renals, J. Carletta, and J. D. Moore. Incorporating speaker and discourse features into speech summarization. In R. C. Moore, J. A. Bilmes, J. Chu-Carroll, and M. Sanderson, editors, *HLT-NAACL*. The Association for Computational Linguistics, 2006.

[13] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[14] A. Stoicke, K. Ries, N. Coccaro, E. Shriberg, R. Bates, D. Jurafsky, P. Taylor, R. Martin, C. V. Ess-Dykema, and M. Meteer. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–373, 2000.

[15] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas. Sentiment in short strength detection informal text. *Journal of the American Society for Information Science and Technology*, 61(12):2544–2558, 2010.

[16] G. Xu and W.-Y. Ma. Building implicit links from content for forum search. In *SIGIR 2006*, pages 300–307, 2006.

[17] X. Xue, J. Jeon, and W. B. Croft. Retrieval models for question and answer archives. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 475–482, New York, NY, USA, 2008. ACM.

[18] J.-M. Yang, R. Cai, Y. Wang, J. Zhu, L. Zhang, and W.-Y. Ma. Incorporating site-level knowledge to extract structured data from web forums. In *WWW '09:*, pages 181–190, New York, NY, USA, 2009. ACM.

[19] L. Zhou and E. H. Hovy. On the summarization of dynamically introduced information: Online discussions and blogs. In *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, page 237. AAAI, 2006.